# Intelligent Image Labeling System for Recognizing Traffic Violations

Dmitriy Titarev [1], Dmitriy Korostelyov [1], Valentin Titarev [2] and Dmitriy Kopeliovich [1]

[1] *Bryansk State Technical University, 7, 50 let Oktyabrya blvd., Bryansk, 241035, Russian Federation*
[2] *Bryansk City Lyceum No. 2, 6, 22 Congress of the CPSU st., Bryansk, 241035, Russian Federation*

**Abstract**

The article examines the problems of traffic violations and possible changes in the city, including in the road infrastructure, based on their analysis. A conclusion is made about the applicability of machine learning methods for preparing and marking images for solving the problem. The article describes in detail the algorithms for automatic marking of images for recognizing traffic violations in order to create a comfortable urban environment. The existing information systems that solve this problem are analyzed, with an indication of their strengths and weaknesses. The description of an intelligent system developed by the authors and combining manual and automatic object recognition is given. The system development tools are described, including the libraries used. The experimental part contains the results of testing the system, incl. neural network training. Information on the number of images and objects on them is given, as well as information on the percentage of correctly detected objects for the automatic image labeling.

**Keywords**

Automatic image labeling, intelligent system, neural networks, OpenCV, traffic violations, YOLO convolutional neural network, comfortable urban environment.

## 1. Introduction

The development of technologies and algorithms for artificial intelligence makes it possible to find new opportunities for its effective use. One of the demanded and promising areas of artificial intelligence is the classification and identification of problem situations resulting from the analysis of images extracted from the video stream.

Separately, the use of computer vision in transport should be highlighted. Already now, violations of traffic rules (hereinafter - TR) are automatically recorded in terms of speed limits, driving at a red traffic light, stopping in the wrong places, crossing the stop line [1, 2]. The consumer of these data systems is the traffic police, and they serve as the basis for issuing fines.

However, collecting statistics on systematic TR violations can become the basis for creating a comfortable urban environment. The project of creating a comfortable environment is one of the priorities in our country and is supported at the level of the Government of the Russian Federation.

For example, pedestrians regularly cross the road in the wrong place. After analyzing the information, we can conclude that they are crossing the road, since there is a public transport stop opposite, and the nearest pedestrian crossing is 500 meters away, and pedestrians are forced to violate TR. The solution to the problem within the framework of the formation of a comfortable urban environment can be the transfer or creation of a new pedestrian crossing.

In order for artificial intelligence systems to successfully cope with such tasks, methods based on machine learning are usually used. One of the main stages in machine learning is the preparation and

*GraphiCon 2021: 31st International Conference on Computer Graphics and Vision, September 27-30, 2021, Nizhny Novgorod, Russia*
EMAIL: titaryovdv@mail.ru (D. Titarev); nigm85@mail.ru (D. Korostelyov); titarev-valentin@mail.ru (V. Titarev); dkopeliovich@rambler.ru (D. Kopeliovich)
ORCID: 0000-0001-5502-2037 (D. Titarev); 0000-0002-0853-7940 (D. Korostelyov); 0000-0001-9867-9848 (V. Titarev); 0000-0003-4095-7029 (D. Kopeliovich)

images labeling for which training is carried out in the future. This stage can take up an important part of the entire machine learning time (tens of percent of the project) [3].

Currently, there are several main approaches to the labeling and classification of images for further preparation [4, 5]: manual, automated, and automatic. In manual markup, the following methods are distinguished: in-house markup by analysts (in-house), outsourcing (attracting a third-party team of analysts), crowdsourcing (attracting individual specialists with the help of specialized platforms). In automatic marking, a synthetic method is distinguished (it involves the generation of new data with given attributes based on generative adversarial networks - GAN) and a software method (it involves the use of automatic marking systems). Automated methods include a combination of software labeling with its further correction and verification by experts. The advantage of manual methods over software ones usually lies in the quality of the labeling, at the same time, the use of software methods of labeling can significantly increase its speed. A logical consequence of these features is the combination of manual and automatic methods and the transition to automated labeling, i.e. use of automatic methods for primary labeling of images with their further correction by experts. The successful application of this approach is based on the use of a convenient and flexible tool for the expert, which allows you to quickly correct the automatically generated image labeling.

Currently, the most common image markup systems are [6]: MakeSense.ai [7], LabelImg [8], VGG Image Annotator [9], LabelMe [10], Scalable [11], RectLabel [12]. Some of these systems support the possibility of automatic primary image labeling using predefined methods and models. This certainly increases the speed of labeling but does not allow using more accurate models for specific tasks, which would show the best results of automatic images labeling. For images labeling, the specialized formats COCO, Pascal VOC, and YOLO are usually used [4]. They have quite wide opportunities and in varying degrees support the following labeling types [4]: Bounding Boxes, Polygonal Segmentation, Semantic Segmentation, 3D Cuboids, Key-Point and Landmark, Lines and Splines. At the same time, the COCO format allows labeling for object detection, keypoint detection, stuff segmentation, panoptic segmentation, and image captioning. However, not all of these formats are fully supported in all of the above systems, which somewhat limits their applicability.

Let us consider in more detail the functionality of existing image labeling systems and the limitations in their use to solve the problem. One of the main criteria is the ability to use images already marked up with the help of a neural network or experts.

**MakeSense.AI.** One of the advantages of this solution is the ability to use an already labeled image as an input parameter. The use of neural networks is limited to only two options: React and React duo, which reduces the using area of the software. Another disadvantage is the low accuracy of object detection.

**LabelImg.** This solution provides an expert with a wide range of tools for selecting objects for further training the neural network while showing good detection accuracy. At the same time, regardless of the complexity of the exposure, the speed of object detection is low. This solution does not allow choosing a neural network for training and loading already labeled images.

**VGG image annotator.** Specialists using this application can recognize objects not only in images but also in video sequences, which significantly increases the scope of its application. It should also be noted that there are convenient tools for selecting objects. This solution is browser-based, which directly affected the speed of object detection. The choice of a neural network and the ability to upload labeled images are missing.

**LabelMe.** The application shows a high speed of operation and good accuracy of object detection for exposures of varying degrees of complexity. An expert can choose a neural network, or combine several neural networks in order to improve the accuracy of object definition. The negative aspects of the application are high system requirements for equipment, high time costs for reading instructions and training, due to the complex interface. In addition, there is no possibility to upload already labeled images.

**Scalable.** This application is designed to work on the Android mobile platform. It supports real-time object detection. The choice of the application implementation option affected the accuracy and speed of its operation, especially in cases of complex exposures. The application lacks the ability to select objects for further training the neural network. The choice of a neural network and the ability to upload labeled images are missing.

**RectLabel.** The solution shows high detection accuracy, providing the expert with convenient tools for working with objects. At the same time, there is no possibility of choosing a neural network, as well as upload already labeled images for further work with them.

In addition to those listed above, we can additionally highlight systems that allow automatic image annotation: V7 Darwin, Deepen.AI, Heartex, Alegion Control, Hasty.ai. Let's consider their strengths and weaknesses.

**V7 Darwin.** The positive aspects of this solution include the high speed and accuracy of object detection, in addition to the automatic annotation mode, there are also tools for manual image labeling, a large number of dataset formats. Among the shortcomings, one should highlight possible problems for users using devices from AMD, as well as high requirements for hardware [13].

**Deepen.AI.** This platform is one of the few that offers the option of working with 3D LiDar, providing the user with a wide range of tools. Deepen.AI has the ability to merge sensor readings (for example, LiDar) and photo / video series. The disadvantages of this solution include high requirements for user qualifications, including knowledge and experience of working with 3D editors [14].

**Heartex.** This solution provides users with open source code, allows you to work with various types of files, is quite easy to learn, and while in the case of using the free version, its functionality is severely limited [15].

**Alegion Control.** This solution allows you to work with video in 4K resolution, including 3D annotation of objects, while imposing very high demands on the hardware [16].

**Hasty.ai.** From the positive aspects of this software, one should highlight the high speed and accuracy of work, an easy-to-learn interface, and support for vector and pixel annotations. However, it does not allow the user to select or load another neural network [17].

These circumstances indicate the relevance of the development of flexible intelligent image labeling systems that allow using various models of automatic primary labeling as well as supporting various labeling formats. A description of the algorithms and software implementation of one of such intelligent systems, namely, an image labeling system for recognizing traffic violations, is given below.

## 2. Analysis of algorithms for automatic image labeling for recognizing traffic violations

Automatic image labeling for recognizing traffic violations is based on algorithms for recognizing traffic objects. The sequence of image flow analysis usually consists of the following stages: image acquisition, automatic detection of objects using a neural network, obtaining an array of enclosing boxes and calculating the characteristics of objects.

Cameras installed at intersections, near roads, traffic lights, pedestrian crossings, etc. transmit an image or video stream to an intelligent system. They are then analyzed using a neural network, with the result that each object must be defined and highlighted using enclosing boxes. The detected objects are transferred to the traffic violations analysis subsystem.

The object recognition algorithm from the received image stream determines all objects present on it, taking into account the probability of their presence, and displays the corresponding labels. For example, as a result of the operation of the algorithm, the presence of an object in the image - "car" with a probability of 0.98 can be determined.

Recognition algorithms not only determine what objects are present in the image but also highlight their boundaries in the form of frames of a certain height and width. This type of labeling is supported by all the formats mentioned above.

To detect an object, we must select areas of the analyzed image and apply a recognition algorithm to it. The location of an object is determined by the location of image areas where the probability of its finding is high. One of the first algorithms for objects detection were algorithms based on features: HOG-algorithm and Haar cascades [18].

### 2.1. Histogram of Oriented Gradients (HOG-algorithm)

The HOG-algorithm (Histogram of Oriented Gradients) is based on the assertion that the shape and appearance of an object in an image can be accurately described by the distribution of pixel intensity gradients corresponding to a given part. A gradient is an approximation by some function of intensity or brightness, the values of which are known only in pixels [19].

When using the HOG algorithm, it is necessary to apply a filter to the color and brightness components (1).

$$[-1 \quad 0 \quad 1] \text{ and } [-1 \quad 0 \quad 1]^T \tag{1}$$

The term HOG-descriptor is directly related to the concept of a block - a rectangular area of image pixels of specified sizes. The block is the main component of the HOG descriptor. A block is a collection of cells that are a collection of pixels.

The next step requires block normalization, for example, L2-norm (2).

$$f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}} \tag{2}$$

where $v$ – unnormalized vector, $e$ – constant.

The HOG-algorithm has several disadvantages:

- the algorithm works well for one or two classes of objects, its efficiency drops sharply when adding objects of a new type;
- low speed of work;
- the accuracy of the algorithm is highly dependent on the type of object being recognized.

## 2.2. Haar Cascades

Haar cascades are digital image features used in pattern recognition. Algorithms that work only with image intensity have great computational complexity. They were used in the first real-time face detector.

Haar cascades are composed of contiguous rectangular regions. First, they are positioned on the image, then the intensity of the pixels in the regions is summed up and the difference of the sums is calculated. This difference represents the value of the feature under consideration, of a given size, located in a certain way on the image.

Consider images of human faces. A common feature for them is that the area around the cheeks is lighter than the area around the eyes. Thus, a common Haar feature for faces is two adjacent rectangular regions lying on the cheeks and eyes.

The main disadvantage of this algorithm is that Haar cascades require a large number of features to accurately describe an object since they are poorly suited for training and classification.

The main advantage of the Haar cascade method in comparison with analogs is speed. But still, the speed of this algorithm significantly decreases with an increase in the number of objects of various types.

## 2.3. Convolutional neural networks

Currently, most problems in the field of computer vision are solved using convolutional neural networks (Convolutional neural networks). One of the first convolutional neural network architectures was R-CNN [20]. It was developed by the UC Berkley team and used to solve the object definition problem.

To improve the performance of this solution, regions that most likely contain objects, rather than a complete image, were fed to the input of the neural network. Regions, in turn, were prepared in advance using a different algorithm. CaffeNet was used as a convolutional neural network.

The process of defining objects using R-CNN can be divided into the following steps:

- determination of image regions using the Selective Search algorithm;
- scaling the region to the size with which the neural network CaffeNet can work;
- obtaining a vector of object features using a neural network;
- carrying out classifications of each feature vector using SVM;

- linear regression of the parameters of the region frame for a more accurate definition of the object.

A convolutional neural network consists of several layers: convolution and activation. The convolution layer processes each previous layer fragment by fragment, while in the process of training the neural network, the coefficients of the convolution kernel are determined.

The convolution results are processed using a non-linear activation function. Typically, this is rectified linear unit – ReLU (3).

$$f(x) = \max(0; x), \quad (3)$$

Using the ReLU has significantly increased the learning rate compared to previously used nonlinear functions such as sigmoids.

This method is faster and more accurate than those described earlier, but at the same time, it places high demands on the amount of disk space for storing a significant number of features. In addition, the weak link of R-CNN is the selection of candidate regions before starting the main algorithm.

## 2.4. You Only Look Once

You Only Look Once (hereinafter - YOLO) is one of the varieties of a convolutional neural network designed to recognize multiple objects in an image. This architecture of the convolutional neural network is currently one of the most popular.

Unlike other algorithms, YOLO applies the neural network once to the entire image as a whole. In this case, a grid is superimposed on the image, then the neural network predicts the boundary frames and the probability that the object being determined will be located in them. YOLO has a higher operating speed and accuracy of object identification compared to R-CNN [21].

This became possible because YOLO unified and combined all the components necessary for detecting objects, and takes into account the context when working with incoming images.

With YOLO you can recognize objects in real-time. Features of YOLO allow you to use it not only on high-power servers and portable equipment but also on mobile devices, which greatly expands the scope of its application.

# 3. Description of the software implementation of the intelligent system

In the developed intelligent system, a convolutional neural network - YOLO was used, which has proven itself most well for solving this class of problems, since when determining traffic violations, a large number of objects of different types can be present in the image.

The software package was developed using the C++ programming language and the OpenCV library. The system interface is shown in Figure 1.

One of the features of the developed intelligent system is the ability to combine manual and automated labeling (Figure 2). At the same time, it is possible to select and plugin with the help of external modules various algorithms for automatic labeling, and the resulting file with markup itself can again enter the system input, which allows the determination of specific types of objects using the most efficient algorithms to detect them.

Also, a distinctive feature of the system is not only the ability to highlight objects in images but also to indicate the belonging of each image to a specific class. For example, you can indicate that the image does not contain signs of traffic violations, or that the image contains signs of a specific violation (people cross a road in the wrong place, improper parking of the car, etc.). This circumstance significantly expands the possibilities of using the developed intelligent system, since not only machine learning methods can be applied to the resulting markup results, but also other methods of intelligent analysis (for example, classification using clustering or building decision trees).

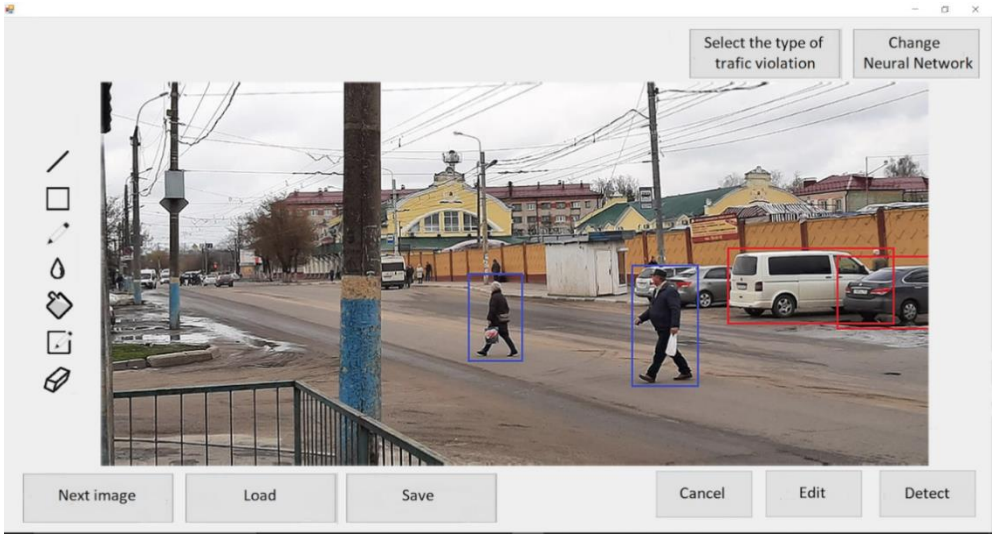


**Figure 1**: Intelligent system interface

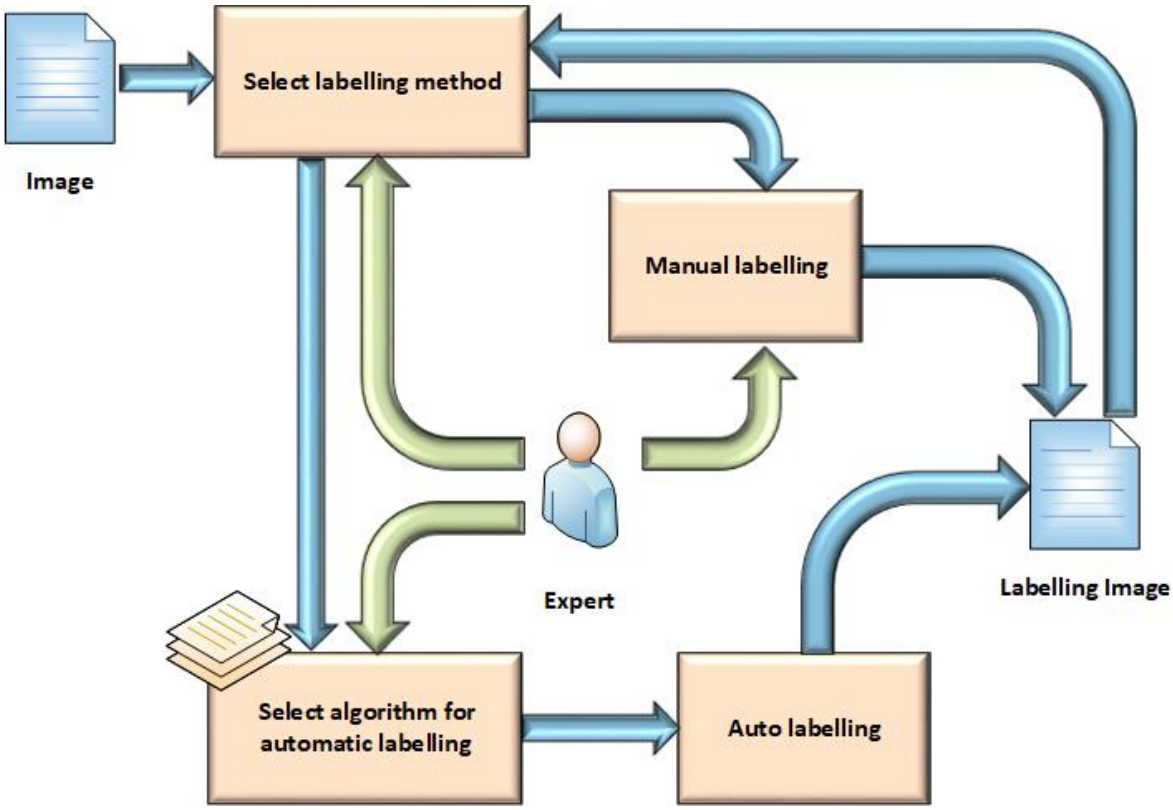


**Figure 2**: Scheme of interaction between an expert and an intelligent system

## 4. Experiment Description

To carry out the experiments, pre-selected images from the Internet were supplemented with a collection of images obtained independently by the authors on the streets of the city of Bryansk.

The use of ready-made trained neural networks showed a fairly significant number of defects in automatic marking. In fig. 3-5 shows examples of traffic objects detection of various types using the YOLO v.3 algorithm based on images obtained on their own.

**Figure 3**: Traffic objects detection at an intersection

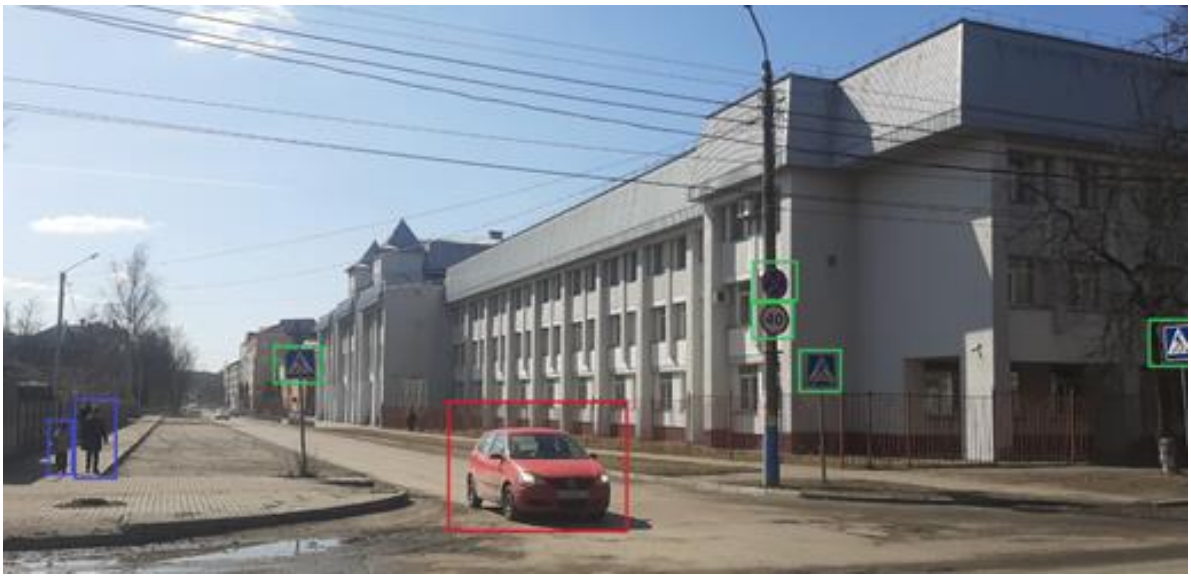

**Figure 4**: Detection of road traffic objects

**Figure 5**: An example of traffic objects detection

As can be seen from the above figures, the recognition quality is not always ideal; therefore, at the first stage of the experiments, the task of training a specialized neural network was set. When implementing the intelligent system, a neural network training approach with a teacher was used. To train the YOLO convolutional neural network, you need a large number of photographs containing objects of various types. To train the neural network, both ready-made photographic objects from the Caltech Pedestrian Dataset database and photographs taken on the streets of Bryansk were used.

In Figure 6, the following objects are visible: a car and road signs (speed limit, stopping and parking are prohibited, as well as information about how long it is valid).


**Figure 6**: Image labeling for neural network training

Knowing the correct result, we can train our neural network. Figure 7 shows a more complex composition of objects in the image, it contains: road signs, cars, and pedestrians.


**Figure 7**: Image labeling for training a neural network on a complex composition

As can be seen from Figure 7, further training of the neural network is required, since not all objects were detected by it. The white minibus at the stop was not detected and was not framed.

The obtained intermediate results showed an improvement in the quality of automatic marking of objects in the image.

Table 1 shows the metrics of training a neural network. From Table 1 that the number of errors after additional training of the neural network during object detection decreased by 6%.

**Table 1**
Neural network training metrics

| Parameter | Value |
|---|---|
| The number of processed images during the first iteration of neural network training | 825 |
| The number of objects in images to be detected | 4847 |
| The number of incorrectly objects detection at the first training iteration | 963 |
| Mean average precision at the first training iteration | 0.985 |
| Mean average recall at the first training iteration | 0.474 |
| The number of processed images after manual labeling of objects by an expert and additional training of the neural network | 758 |
| The number of objects in images to be detected in the second iteration | 4296 |
| The number of incorrectly objects detection in the second iteration | 604 |
| Mean average precision in the second iteration | 1 |
| Mean average recall in the second iteration | 0.715 |

Further, a number of rules were developed for automatic classification of the type of image (determining the presence of traffic violations and the type of traffic violations), which were integrated into the intelligent system. Depending on the presence of one segment of the image of objects of different types, the following classes of rules were distinguished:

1. On one segment of the image, there are objects of different types. For example:

- Pedestrian crossing sign, car, people. Possible violation - the driver of the car did not let pedestrians pass at the pedestrian crossing.
- Cars, people. Possible violation - pedestrians cross the road in the wrong place.
- Traffic light, car, people. Possible violation - the driver of the car did not let pedestrians pass or pedestrians cross the road at a traffic light prohibiting sign.

2. The imposition of one object on another. For example:

- One of the objects is road markings (solid or double line), the other is a car. A possible violation is the intersection of a solid or double solid line.
- One of the objects is a designated lane for cyclists, the other is a car. Potential violation - driving a car into a designated lane for cyclists.

The application of these rules added the ability for an expert to receive an automatic hint when classifying the types of violations in images.

## 5. Discussion of Experimental Results

The above results of experiments on automatic image labeling based on neural networks have shown good results, but they cannot be recognized as ideal (even after additional training of neural networks). It is for this reason that one cannot rely solely on automatic methods for high-quality image labeling, because in this case there is a high risk of missing essential details or, conversely, finding missing objects. This is especially important for the automatic detection of traffic violations. Therefore, the involvement of experts in the image labeling procedure is a demanded and important approach.

The combination of these approaches in the developed intellectual system made it possible, on the one hand, to significantly reduce the time of primary marking of the image due to the automatic detection of objects and classification of types of violations. On the other hand, it provided the expert

with the opportunity to make changes manually, as well as to choose different methods of automatic detection (different neural networks, different modules) depending on the types of objects being detected, and made it possible to significantly improve the quality of the resulting labeling.

## 6. Conclusion

The development of an intelligent image labeling system for recognizing traffic violations allows them to be used for further analysis in order to create a comfortable urban environment. At the same time, the use of machine learning methods for preparing and images labeling has significantly reduced the operating time of the entire system as a whole and provided the experts involved in the analysis with a convenient, multifunctional tool.

With the combination of manual and automatic marking of images, the ability to use already labeled images as an input parameter has significantly reduced the number of errors in object detection using a neural network.

Equally important is the ability to indicate the type of traffic violation or the choice of the neural network used in the intelligent system, which ultimately also affected the speed of its operation and the number of errors in the resulting data.

Integration with information systems containing information about road signs and road labeling is a promising method for improving the quality of automatic preliminary labeling of road objects.

Possible directions for further development of the system are:

- creation of a universal platform suitable for labeling and classifying images for various tasks;
- creation of a multi-user system for parallelizing the labeling procedure;
- support for cloud storage of images for organizing centralized access to it for various experts.

## 7. References


[1] Fozia Mehboob. "Mathematical model based traffic violations identification" Computational and Mathematical Organization Theory. 2019. P. 302-318. doi: 10.1007/s10588-018-9264-x.

[2] Shiva Asadianfam. "TVD-MRDL: traffic violation detection system using MapReduce-based deep learning for large-scale data" Multimedia Tools and Applications. 2021. P. 2489-2516. doi: 10.1007/s11042-020-09714-8.

[3] Data Engineering, Preparation, and Labeling for AI 2019. URL: https://www.cloudfactory.com/reports/data-engineering-preparation-labeling-for-ai.

[4] 5 Approaches to Data Labeling for Machine Learning Projects. URL: https://lionbridge.ai/articles/5-approaches-to-data-labeling-for-machine-learning-projects/.

[5] A. Zakharova and D. Korostelyov, "Visual Classification of Data Sets with the Assistance of Experts in the Problems of Intelligent Agents Learning for Incompletely Automated Control Systems," 2019 Dynamics of Systems, Mechanisms and Machines (Dynamics), 2019, pp. 1-5, doi: 10.1109/Dynamics47113.2019.8944638.

[6] Image Data Labelling and Annotation - Everything you need to know. URL: https://www.xailient.com/post/image-data-labelling-and-annotation.

[7] Make Sense. URL: https://www.makesense.ai.

[8] GitHub – tzutalin/labelImg: LabelImg is a graphical image annotation tool and label object bounding boxes in images. URL: https://github.com/tzutalin/labelImg.

[9] VGG Image Annotator: a standalone image annotator application packaged as a single HTML file that runs on most modern web browsers. URL: https://gitlab.com/vgg/via.

[10] LabelMe, the open annotation tool. URL: http://labelme.csail.mit.edu/Release3.0/.

[11] Scalabel. URL: https://scalabel.ai/.

[12] RectLabel – Labeling images for bounding box object detection and segmentation. URL: https://rectlabel.com/.

[13] V7 - AI Data Platform for ML Teams. URL: https://www.v7labs.com/.

[14] Industry leading multi-sensor, LiDAR annotation and labelling tools. URL: https://www.deepen.ai/.

[15] Data Labeling Platform for Machine Learning – Heartex. URL: https://www.heartex.com/.
[16] Alegion | Data Labeling Software Platform. URL: https://www.alegion.com.
[17] Hasty.ai - A single application for all your vision AI needs. URL: https://www.hasty.ai.
[18] S. Ren, K. He, R. Girshick, and J. Sun. "Faster R-CNN: To-wards real-time object detection with region proposal net-works." IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017. Vol. 39(6). P. 1137-1149. doi: 10.1109/TPAMI.2016.2577031.
[19] Navneet, D. "Histograms of Oriented Gradients for Human Detection" IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVRP). 2005. doi: 10.1109/CVPR.2005.177.
[20] J. Dai, K. He, Y. Li, S. Ren, and J. Sun. "Instance-sensitive fully convolutional networks" European Conference on Computer Vision. 2016. P. 534-549. doi: 10.1007/978-3-319-46466-4_32.
[21] Redmon, Joseph and Ali Farhadi. "YOLOv3: An Incremental Improvement." ArXiv abs/1804.02767 (2018): n. pag.